ISBN 0 7334 0034 5

A COMPUTER CODE FOR ENCLOSED NATURAL CONVECTION IN THE BOUNDARY LAYER REGIME

N.A.Kelson

Report 1990/FMT/2 May 1990

School of Mechanical and Manufacturing Engineering,
University of New South Wales,
P.O. Box 1, Kensington, N.S.W., Australia 2033

Abstract

A computer code is developed for the numerical prediction of natural convection in rectangular two-dimensional cavities at high Rayleigh numbers. The governing equations are retained in the primitive variable form. The numerical method is based on finite differences and an ADI scheme. Convective terms may be approximated with either central or hybrid differencing for greater stability. A non-uniform grid distribution is possible for greater efficiency. The pressure is dealt with via a SIMPLE type algorithm and the use of a fast elliptic solver for the solenoidal velocity correction field significantly reduces computing times. Preliminary results indicate that the code is reasonably accurate, robust and fast compared with existing benchmarks and finite difference based codes, particularly at high Rayleigh numbers. Extension to three-dimensional problems and turbulence studies in similar geometries is readily possible and indicated.

CONTENTS

1 Introduction 1

2 Mathematical Formulation 2

2.1 Governing Equations 2

2.2 Boundary Conditions 3

2.3 Non-dimensionalisation 3

3 Numerical Method 4

3.1 Discretization 4

3.2 Solution procedure 6

3.3 Application of ADI 6

3.4 Solution of the Velocity Corrector Field 8

3.5 Implementation of the Boundary Conditions 9

4 Program Structure 11

5 Sample Results and Discussion 11

6 Acknowledgements 16

7 References 16

Figures 19

NOMENCLATURE

A_r	cavity aspect ratio, H/L
g	magnitude of gravitational acceleration
Gr	Grashof number, $g\beta(T_h - T_c)L^3/\nu^2 = Ra/Pr$
H	height of cavity
L	width of cavity
Nu	Nusselt number
P	pressure
Pr	Prandtl number, ν/α
Ra	Rayleigh number, $g\beta(T_h - T_c)L^3/\nu\alpha$
Sr	ratio of time and space steps, $\Delta t = Sr * \{min(\Delta x, \Delta y)\}^2$
t	time
T	temperature
T_c	temperature of isothermal cold wall
T_h	temperature of isothermal hot wall
T_{ref}	reference temperature, $(T_h + T_c)/2$
u, v	x, y velocity components
U_c	convection velocity, $\nu Gr^{\frac{1}{2}}/L$
x, y	co-ordinate directions
α	thermal diffusivity
β	volumetric expansion coefficient
ν	kinematic viscosity
ρ	density
ϕ	velocity corrector

1 Introduction

Natural convection in enclosures has been extensively studied numerically due to its applicability to a wide range of practical problems. These include solar energy collection, double glazing and air currents in rooms, crystal growth in liquids and natural convection cooling of electronic components. The wide range of physical geometries and fluid properties ensure that many flow regimes are possible: steady or unsteady, laminar or turbulent, two or three dimensional. Relevant reviews covering different aspects have been given by Catton 1978, Ostrach 1982b, Hoogendorn 1986, Simpkins 1987 and Ostrach 1988.

In order to better understand the physical processes underlying a particular application it is often convenient to consider simplified geometries and idealised boundary conditions. To this end, we consider here a Boussinesq fluid enclosed in a rectangular two dimensional cavity with differentially heated isothermal vertical walls and adiabatic horizontal walls. An analysis of this standard problem depends on three independent parameters, namely, the Rayleigh and Prandtl numbers and the Aspect Ratio (Batchelor 1954).

In this report we examine the case of air ($Pr = 0.71$) contained in a square cavity ($A_r = 1$) and Rayleigh numbers for which the flow is laminar. In particular, a numerical method is developed suitable for high Ra, where the flow is characterised by thin thermal and hydrodynamic boundary layers next to the vertical walls and a quiescent, stratified core region where the vertical temperature gradient is approximately constant and the velocites are very small (Eckert and Carlson 1961, Elder 1965a). This boundary layer regime may be identified for all Rayleigh numbers greater than 10^4, although the flow is initially steady, then unsteady, and finally turbulent as Ra is increased. The turbulent regime appears to begin at $Gr \approx 10^9$ (Elder 1965b, Turner 1973, Lankhorst et al 1989). The beginning of the unsteady laminar regime is not well known, and is usually assumed to occur about a decade before the onset of turbulence. In general the problem is analytically intractable and numerical methods of solution must be employed.

The development of an acceptable numerical method for enclosed natural convection studies at high Rayleigh number is an ongoing research area where only partly-clear answers have emerged thus far. Typical numerical problems which may have to be considered include choosing a suitable non-dimensionalisation and discretization, overcoming the rapidly increasing stiffness of the governing equations, and properly resolving the thin boundary layers. Other considerations include the relative accuracy of the numerical solutions so obtained and the computing time and memory requirements. In the work presented here, the development of a computer code which attempts to overcome some of these problems is given.

2 Mathematical Formulation

2.1 Governing Equations

The equations governing the natural convection of a viscous Newtonian fluid are well known (Gebhart et al 1988). These equations express the conservation of mass, momentum and energy and are presented below for a constant property, Boussinesq fluid where the pressure work and viscous dissipation are neglected and the flow is assumed to be two dimensional.

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{1}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{1}{\rho}\frac{\partial P}{\partial x} + \nu\nabla^2 u \tag{2}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{1}{\rho}\frac{\partial P}{\partial y} + \nu\nabla^2 v + g\beta(T - T_{ref}) \tag{3}$$

$$\frac{\partial T}{\partial t} + u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} = \alpha\nabla^2 T \tag{4}$$

The pressure gradient term in equations 2 and 3 is that which arises due to the fluid motion only.

The equations 1, 2, 3 and 4 represent a non-linear, coupled, partial differential set and the solutions may be locally parabolic (i.e. transient), elliptic (i.e. steady) or hyperbolic (i.e. wave-like) depending on the physical situation being considered.

An alternative form of the governing equations is often used which eliminates the pressure as a dependent variable. This is the vorticity-stream function (for two dimensional) and vorticity-vector potential (for three dimensional) formulations. The vorticity transport equation is obtained by cross-differentiation and subtraction of the momentum equations 2 and 3. Some advantages of this approach are that the continuity constraint (equation 1) is identically satisfied, the pressure is eliminated as a dependent variable, and, for two dimensional studies, the number of equations is reduced from four to three. Disadvantages include the imposition and implementation of boundary conditions for the vorticity, and the extension from two to three dimensional flow. In particular, the treatment of three dimensional and multiply-connected regions in this formulation appears to have some unresolved difficulties (Reizes et al 1984, Wong and Reizes 1986). Nevertheless, the choice of one formulation over another is neither clear-cut nor decisive and one must usually evaluate the merits or otherwise of the final choice with reference to the specific problem being considered.

In this work, we prefer to retain the equations in the primitive variable form, since the boundary conditions are more readily imposed and extension to turbulence and three dimensional studies is less cumbersome.

2.2 Boundary Conditions

For the standard problem, the boundary conditions for the velocity are the no-slip and no-throughflow conditions on all walls.

no-throughflow:

$$u(x = 0, y) = u(x = L, y) = 0 \tag{5}$$

$$v(x, y = 0) = v(x, y = H) = 0 \tag{6}$$

no-slip:

$$v(x = 0, y) = v(x = L, y) = 0 \tag{7}$$

$$u(x, y = 0) = u(x, y = H) = 0 \tag{8}$$

For the temperature, the two vertical walls are isothermal and the horizontal walls are adiabatic:

$$T(x = 0, y) = T_h \ , \ T(x = L, y) = T_c \tag{9}$$

$$\frac{\partial T}{\partial y}(x, y = 0) = \frac{\partial T}{\partial y}(x, y = H) = 0 \tag{10}$$

2.3 Non-dimensionalisation

The non-dimensionalisation of the governing equations has been the subject of some debate in the literature. It appears that the correct choice of scaling factors can lead to computational advantages and enhanced physical understanding (Ostrach 1982a, de Vahl Davis 1986).

For this study, the convection velocity (U_c) arises naturally from simple physical considerations (Gebhart et al 1988), and appears to be the most appropriate choice for high Ra computations (Ostrach 1982a,1982b). Using $(T_h - T_c)$, U_c and L as scale factors, the dimensional equations may be transformed into their non-dimensional form. For sake of brevity, new symbols are not defined for the dimensionless variables. e.g. the non-dimensional form of equations 3 and 4 are

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{\partial P}{\partial y} + \frac{1}{\sqrt{Gr}}\nabla^2 v + \frac{1}{2}T \tag{11}$$

$$\frac{\partial T}{\partial t} + u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} = \frac{1}{Pr\sqrt{Gr}}\nabla^2 T. \tag{12}$$

Similarly, the boundary conditions may be rewritten, e.g. equations 6 and 9 become

$$v(x, y = 0) = v(x, y = A_r) = 0 \tag{13}$$

$$T(x = 0, y) = -T(x = 1, y) = 1. \tag{14}$$

All quantities in equations 11, 12, 13 and 14 are dimensionless.

3 Numerical Method

3.1 Discretization

The non-dimensionalised governing equations were discretized using finite differences on a non-uniform, staggered grid. Figure 1 shows the exact definition of the computed variables at the various staggered grid locations and should be referred to when deriving the difference equations. One should note that the computed velocities are located exactly midway between the grid lines and that the momentum equations will be discretized at these mid-points.

Using the mesh and variable definitions above, the development of a discretization scheme which involves only nearest neighbours is now given. An implicit coupling (Crank-Nicholson has been used) of only nearest neighbours will ensure that the resulting matrices will be tridiagonal so that the efficient Thomas algorithm and an Alternating Direction Implicit (ADI) procedure can be adopted.

If the mesh distribution is uniform then a natural choice for the discretization of the spatial derivatives is the formally second order accurate central differencing scheme. If, on the other hand, the more general non-uniform mesh of Figure 1 is used, then it is important to note that any discretization of the diffusion terms which employs only nearest neighbours will be at best only first order accurate. This may be verified by straightforward Taylor Series expansions. This means that the choice of discretization for both the convective and diffusive terms becomes somewhat arbitrary, although we should ensure that central differencing is recovered when a uniform grid is used.

Indeed, one should not necessarily choose a scheme which minimises the order or magnitude of the formal truncation error: robustness, ease of programming, physical realism and other factors may have to be considered (Patankar 1980).

In the present work, a discretization scheme has been chosen based on simplicity and minimisation of computational cost. These factors are particularly important on a staggered mesh where additional averaging operations are always required when solving for the staggered variables compared with the non-staggered ones.

Consider a general variable $\xi^k_{m,n}$ whose subscripts m,n denote the location of the grid point in the x, y directions and superscript k denotes time level. ξ may be staggered or non-staggered. The following analysis should be applied when obtaining finite difference approximations for ξ and its derivatives in the momentum and energy equations.

The derivative of ξ in x direction is approximated by the mean of its forward and backward differences. Using standard difference operator notation we have

$$\frac{\partial \xi^k_{m,n}}{\partial x} \simeq \frac{1}{2}(\Delta^+_x + \Delta^-_x)\xi^k_{m,n}. \tag{15}$$

This expression is first order accurate, and it is in fact possible to construct a formally second order accurate expression for the first partial derivative by using the geometric rather than arithmetic mean in equation 15. The reason that we do *not* adopt the second order expression is because in practice it was found to perform no better than its first order counterpart on typical grids. This paradoxical result has been resolved theoretically (Mautenffel and White 1986, Kreiss et al 1986) for one-dimensional problems.

The second order partial derivative can be approximated in like fashion,

$$\frac{\partial^2 \xi^k_{m,n}}{\partial x^2} \simeq (\frac{\Delta^+_x - \Delta^-_x}{d_{m,n}})\xi^k_{m,n} \tag{16}$$

where

$$\begin{aligned} d_{m,n} &= hx(m) \quad \text{if } \xi \text{ is computed at the same location as } u_{m,n}\ , \\ &= \tfrac{1}{2}(hx(m) + hx(m+1)) \quad \text{otherwise}\ . \end{aligned}$$

Equations 15 and 16 reduce to the usual central difference expressions when a uniform grid is used.

First order upwinding may be incorporated by slight modification of equation 15, thus

$$\frac{\partial \xi^k_{m,n}}{\partial x} \simeq \frac{1}{2}[(1 - sgn(u))\Delta^+_x + (1 + sgn(u))\Delta^-_x]\xi^k_{m,n} \tag{17}$$

where

$$sgn(u) = \frac{u^k_{m,n}}{|u^k_{m,n}|}.$$

A further modification is to use the hybrid differencing scheme. In this case "central" differencing (Equation 15) is used until the cell Reynolds or Peclet numbers exceed 2, after which the upwind differencing scheme (Equation 17) is used.

In the work reported here, equation 15 has been used for the convective terms unless otherwise stated.

Expressions for the spatial derivatives in y direction may be obtained in similar fashion.

If any variable in the momentum and energy equations is not defined at a point where it is required, it is evaluated as the mean of the values of this variable at the nearest two or four neighbours.

In the continuity equation the central point for evaluating the finite difference approximations is at the intersection of the grid lines. The divergence of the velocity is approximated there by

$$\nabla.\vec{V} \simeq \frac{u_{m+1,n} - u_{m,n}}{\frac{1}{2}[hx(m) + hx(m+1)]} + \frac{v_{m,n+1} - v_{m,n}}{\frac{1}{2}[hy(n) + hy(n+1)]} \tag{18}$$

3.2 Solution procedure

The solution procedure consists of the following steps:

(i) The velocities at the new time level are predicted from the momentum equations using a standard ADI procedure, described below. These velocities do not in general satisfy the continuity equation.

(ii) The predicted velocities are then corrected via the solution of a Poisson equation for the soleniodal velocity corrector, described below. The corrected velocities in general no longer satisfy the momentum equations.

(iii) The pressure at the new time level can be estimated by either solving a Poisson equation with Neumann boundary conditions or the sum of the pressure at the previous time level and the velocity corrector obtained from the previous step.

(iv) The temperature at the new time level is predicted from the energy equation using a standard ADI procedure.

(v) Steps (i) to (iv) are repeated at each time level. If a final steady state solution is desired, then the procedure is continued until the transients have decayed.

3.3 Application of ADI

The discretized momentum and energy equations may be written in a general form as follows

$$\frac{\partial \xi}{\partial t} = (A_x + A_y)\xi + S \tag{19}$$

where A_x, A_y are matrix operators which represent the linearized finite difference approximations for the convective and diffusive terms in the x and y directions respectively, and S is a finite difference form of the source term. Introducing the Crank-Nicholson scheme into equation 19 and conveniently rewriting it in terms of new variable ω, viz.

On the left hand side, let

$$\omega = \frac{\partial \xi}{\partial t} = \frac{\xi^{k+1} - \xi^{k}}{\Delta t} \tag{20}$$

On the right hand side, replace ξ by

$$\tfrac{1}{2}(\xi^{k+1} + \xi^k)$$

or, in terms of ω defined in equation 20,

$$\xi^k + \tfrac{1}{2}\Delta t\omega \ .$$

The source term may be treated by taking values that are available at the current time level. After making these substitutions and rearranging we obtain the following matrix equation:

$$\omega - \frac{1}{2}\Delta t(A_x + A_y)\omega = (A_x + A_y)\xi^k + S^k. \tag{21}$$

Since the right hand side of equation 21 is completely known, we may now solve for ω and use equation 20 to obtain the value of ξ at the new time level.

Equation 21 may be solved via an ADI scheme such as that proposed by Douglas 1962, which has been also independently given by Brian 1961, and Samarskii and Andreyev 1963, and has been fully described elsewhere (Isenberg and de Vahl Davis 1975):

$$(I - \frac{1}{2}\Delta t A_x)\omega^* = (A_x + A_y)\xi^k + S^k. \tag{22}$$

$$(I - \frac{1}{2}\Delta t A_y)\omega^{**} = \omega^* \tag{23}$$

$$\xi^{k+1} = \xi^k + \Delta t\omega^{**} \tag{24}$$

In equations 22 and 23, I denotes the unit matrix and in equation 24, ω^{**} is an $O(\Delta t)$ approximation to ω defined in equation 20 .

The solution method consists of the following steps:

(i) The solution of equation 22 in the x direction for each row successively using a standard tridiagonal algorithm.

(ii) The solution of equation 23 in the y direction for each column successively using a standard tridiagonal algorithm.

(iii) The solution of equation 24 for each point of the solution field.

The operators A_x and A_y on the left hand side of equations 22 and 23 are interchanged every alternate step to improve the symmetry of the computational procedure.

3.4 Solution of the Velocity Corrector Field

The continuity equation is not in general satisfied after prediction of the velocities at the new time level using the momentum equations.

Following Briley 1974 and Patankar and Spalding 1972, an irrotational velocity corrector ϕ is introduced such that

$$\frac{\partial}{\partial x}(u_p + \frac{\partial \phi}{\partial x}) + \frac{\partial}{\partial y}(v_p + \frac{\partial \phi}{\partial y}) = 0 \tag{25}$$

where u_p and v_p are the velocities obtained at the prediction stage.

Since u_p and v_p are everywhere zero on the boundaries, this leads to a Poisson equation with Neumann boundary conditions for ϕ. i.e.

$$\nabla^2 \phi = -\frac{\partial u_p}{\partial x} - \frac{\partial v_p}{\partial y} \tag{26}$$

with boundary conditions

$$\frac{\partial \phi}{\partial \eta}\Big|_\Gamma = 0 \tag{27}$$

where η is a unit normal to the boundary Γ.

Velocity corrector $\phi_{m,n}$ is located at the intersection of grid lines (see Figure 1) as this is the central point for differencing the right hand side of equation 26.

Usually attention must be given to the discrete form of the integral constraint required when solving Poisson equations with normal derivative boundary conditions, i.e. Green's Theorem: $\int\int(\nabla^2 P)da = \oint(\frac{\partial P}{\partial \eta})dl$. In the present case this may be conveniently ignored, provided that the solution arrays retain the centro-symmetric properties of the continuous solution at each time level.

Equation 26 may be solved by the ADI method outlined previously. Using this method, a number of internal iterations are usually required in order to obtain a good approximation for ϕ at each time level. If insufficient iterations are performed, then a poor estimate for the correction velocity will be obtained. In this case the solution procedure can be slowed considerably, or, worse still, convective instability due to unchecked growth of mass sources may result in lack of convergence (Piacsek and Williams 1970).

Alternatively, a fast direct method may be used. An important restriction is that a suitable implementation of the "staggered" boundary conditions is required since the Neumann condition must be applied at the solid boundary while ϕ is not actually located there.

A two dimensional elliptic solver based on cyclic reduction and fast Fourier transform has been implemented which is able to handle these boundary conditions (Schumann and Sweet 1976). A restriction is that a non-uniform grid in *both* directions is not permitted. This restriction does not turn out to be too severe, since adequate resolution of the boundary layers for most Ra is often

achieved by using a non-uniform grid only in the direction normal to the heated and cooled walls. If a non-uniform grid in both directions is desired then the iterative ADI method of solution for ϕ must be employed.

When implemented properly, the elliptic solver that has been chosen produces a very accurate solution for ϕ which in turn yields corrected velocities which satisfy the continuity equation to order of roundoff at each time level.

This results in two significant advantages: (i) for transient studies a large number of internal iterations is no longer required in order to obtain accurate values for ϕ at each time level; (ii) the overall numerical method is more robust so that large time steps are possible when marching towards a steady solution.

Preliminary computations revealed that the computer time required when using the direct method to solve for ϕ at each time level was equivalent to performing about 4-5 ADI sweeps . In order to get a solution of comparible accuracy at each time level, the number of internal iterations required when using the ADI method was around 50. The use of the exact solver therefore represents a savings of approximately a factor of 10 times for this problem and should be preferred.

3.5 Implementation of the Boundary Conditions

The staggered mesh, first introduced by Harlow and Welsh 1965, is shown near the vicinity of the solid boundary (Γ) in Figure 2. Note that the computed velocity components normal to the wall are located on it, while the pressure (P) and velocity corrector (ϕ) are not. In one of the algorithms considered here such an arrangement allows the pressure at the new time level to be obtained without requiring an explicit prescription for its boundary conditions.

On the other hand, the velocities parallel to the wall (u_2, v_2, etc.) are not located on the physical boundary either, so that some prescription is necessary when computing their associated derivatives adjacent to the wall (i.e. at u_2, v_2, etc.). This usually requires either the use of an external point (u_1, v_1, etc.) or one-sided differencing into the flow domain. Both methods must be derived by incorporating the known values of the velocities on the boundaries (u_Γ, v_Γ, etc.).

The very simplest and lowest order prescription is to use a two point reflection technique:

$$u_1 = 2u_\Gamma - u_2 \tag{28}$$

$$v_1 = 2v_\Gamma - v_2 \tag{29}$$

Substitution of equations 28 and 29 into the finite difference approximations for the first and second partial derivatives at u_2, v_2, etc. result in spatial approximations of first and zeroth order respectively. The latter is particularly unacceptable because it means that at points u_2 and v_2 the resulting difference equation is no longer consistent with the differential equation. Despite this, the implementation of equations 28 and 29 leads to satisfactory results in practice.

While some theoretical reasons for this have been proposed (see Peyret and Taylor 1983 and references therein), a simple physical explanation would appear to be that near the walls viscous effects predominate and result in closely linear velocities. Indeed, initial computations which replaced the no-slip condition by a "linear power law" in Neumann form at the first computational grid point also produced good results, confirming the presence of a viscous sublayer. Apparently, the zeroth order error term was sufficiently small in the cases that were tested ($Ra \leq 10^6$) so as not to lead to an overall loss of accuracy of the numerical solution.

A higher order "reflection technique" based on three points on a *locally* uniform grid is as follows:

$$u_1 = \frac{1}{3}(u_3 - 6u_2 + 8u_\Gamma) \tag{30}$$

$$v_1 = \frac{1}{3}(v_3 - 6v_2 + 8v_\Gamma) \tag{31}$$

Equations 30 and 31 are equivalent to extrapolating the value of the external point u_1 (v_1) by first fitting a parabola through u_Γ, u_2 and u_3 (v_Γ, v_2 and v_3).

When substituted into the finite difference approximations for the first and second derivatives at u_2 and v_2, the use of equations 30 and 31 now result in spatial approximations with truncation error of second and first order respectively, i.e. local consistency is restored.

Experiments have been performed with these and other formulae suitable for the more general situation of a locally non-uniform grid, using both reflection and one sided expressions. A general summary of experience is as follows:

> When using low order expressions like equations 28 and 29 and similar one-sided variants, the local curvature of the solution is underestimated. This results in an over-prediction of near-wall maxima (e.g. velocity) on all grids, while the locations of these maxima are adequately predicted.
>
> The use of higher order expressions in general results in the local curvature of the solution being over-estimated. In this case the near-wall maxima tend to be under-predicted, while their locations are in general better resolved compared with the lower order implementations.
>
> As the order of the expressions is increased more grid points extending into the flow domain need to be incorporated into the analysis. In this case the predictions for both the near-wall maxima and its location quickly deteriorate because e.g. grid points beyond the local boundary layer thickness are included.

In the computations reported here, the "parabolic" expressions of equations 30 and 31 have been used. These provide a locally consistent approximation which appeared to work well for all the cases that have been tested so far.

Some of the points covered in this subsection have been discussed by Peyret and Taylor 1983, to which the reader is referred for additional details.

4 Program Structure

The Fortran program that has been developed is highly structured and fully commented. Subroutines perform single tasks wherever possible and only arrays and constants necessary for the specific task are passed via argument lists or common blocks respectively. The resulting code is highly readable and can be easily modified to solve a variety of related problems in similar geometries. The tilted cavity, the driven cavity or the standard problem with linear temperature boundary conditions are examples which may be solved with relatively minor modifications to the existing program.

The purpose of each subroutine is not described here since this is either obvious or clearly commented after every subroutine statement in the source code. However, two points of clarification may be made.

A separate subroutine call is required for each solution variable since the implementation of a non-uniform staggered grid means that a different discretization and array space is required for each variable. Also, each subroutine which solves for a principle variable is structured in the same way: a subroutine is first called which sets the source array for the current time level, the discretization is then set, a call to the ADI solver is made and, finally, another subroutine call updates the boundary conditions. This allows changes to be made efficiently and in an isolated manner. Figure 3 gives the principle features of the solution procedure.

5 Sample Results and Discussion

In order to assess the relative accuracy and efficiency of the program, the standard problem of natural convection of air in a square two dimensional cavity has been considered. The range of Ra is restricted to those for which a final laminar steady state solution is likely.

The standard problem has been the subject of an international numerical comparison exercise (de Vahl Davis and Jones 1983) and very accurate solutions, suitable as a benchmark, are available for $10^3 \leq Ra \leq 10^6$ (de Vahl Davis 1983).

Of importance also are the recent computations of Behnia et al 1988, who were interested in the relative cost, as well as accuracy, of various methods. They compared a number of finite difference codes based on different vorticity formulations. On the basis of computer time, relative accuracy and memory requirements, it may be concluded that a stream function-vorticity based code (de Vahl Davis 1976) using the method of false transient (Mallinson and de Vahl Davis 1973) and a stable fast marching scheme (Behnia et al 1990) appeared to be the best choice for standard problem considered here.

The benchmark solution of de Vahl Davis 1983, and the results of Behnia et al 1988 provide useful information which we may compare with the present compu-

Table 1. Mean CPU time per iteration per mesh point.

Mesh	CPU sec. per iteration	(per mesh point)
21×21	0.185	(4.63×10^{-4})
41×41	0.735	(4.59×10^{-4})
81×81	3.170	(4.95×10^{-4})
stable $\psi - \zeta$		(1.75×10^{-4})

tations. The data for the stable stream function-vorticity code (denoted stable $\psi-\zeta$) examined by Behnia et al 1988 was chosen for comparison with the computations performed here and these were obtained in double precision on the same computer (IBM 4381) using an identical CPU timing routine and convergence criteria.

In table 1 a measure of the time taken for a single global iteration is given. The time per iteration per mesh point appears to be approximately independent of mesh size. The present primitive variable based code is more than 2.6 times slower than its stream function-vorticity based counterpart in completing a global iteration. This is expected because more equations need to be solved and a larger operation count is required for the staggered variables. Also the implementation of the fast elliptic solver incurs some additional expense.

The total "cost" is given in Table 2 (i.e. the required number of iterations and total CPU time to reach convergence). In all cases the solutions were obtained starting from the initial rest condition $u = v = 0$ and the initial condition for temperature was assumed to be linear.

Also included in Table 2 is the optimum time step (given in terms of Sr) for the primitive variable code, in order to reach steady state in as few global iterations as possible. The values given show that very large time steps are possible, demonstrating the robustness of the code. Especially noteworthy are the large time steps possible at high Rayleigh numbers ($Ra \geq 10^5$). This significantly reduces the required number of global iterations such that for a mesh of 81×81, savings in computer cost of up to 2.5 times ($Ra = 10^5$) and 4.2 times ($Ra = 10^6$) are achieved when compared with the fastest $\psi - \zeta$ code reported by Behnia et al 1988.

Another point is that the number of iterations required using the primitive variable code is virtually independent of the initial conditions for temperature. At time t=0 a step change at the walls has been used, as well as a linear distribution of isotherms parallel to the horizontal walls, with essentially identical results. The latter was thought to be advantageous at high Ra, where the core region is large and the temperature distribution in the core is vertically, linearly stratified. This was not, however, borne out by the computations.

One should also note that for the primitive variable computations, the method of false transient and the stable fast marching scheme were *not* used. Implementation of both these techniques would be expected to reduce even further

Table 2. Optimum time step and no. iterations required to reach steady state (total CPU sec.) for different Ra and mesh sizes.

Mesh	*Ra*	Sr(optimum)	primitive variable	stable $\psi - \zeta$
21 × 21	10^3	1.50	379 (70.1)	62 (4.4)
	10^4	1.75	108 (20.0)	64 (4.6)
	10^5	1.75	52 (9.6)	125 (8.9)
41 × 41	10^3	70.0	824 (606)	246 (67.5)
	10^4	15.0	250 (184)	180 (48.9)
	10^5	4.50	98 (72.0)	175 (47.5)
	10^6	2.125	126 (92.6)	228 (61.9)
81 × 81	10^5	14.0	168 (533)	1145 (1310)
	10^6	8.0	132 (418)	1539 (1754)

the total CPU times for the present computations.

In Table 3 an indication of the accuracy is given by comparing some characteristic features of the solution with a benchmark solution. It should be noted that the maximum values given in Table 3 are mesh point values only, while the benchmark was obtained via iterpolated maxima. The present method evidently provides solutions of comparable accuracy and, in particular, provides good coarse grid solutions.

The results that have been presented so far were obtained on uniformly spaced grids. However, the code that has been developed here does allow non-uniform meshes to be employed. One would expect that the appropriate use of such grids would result in increased resolution in regions where the flow varies appreciably, while at the same time avoiding the computational burden of additional mesh points that the imposition of a uniform grid would require. This basic philosophy is embodied in a number of numerical methods (e.g. coordinate transformations, self adapting grids, multi-griding, etc.) and has been discussed by many authors (e.g. Patankar 1980, Roache 1972, Anderson et al 1984).

An important consideration when using non-uniform grids is the estimation of error of the solution so obtained. This is not straightforward because both magnitude and order of the leading error terms vary locally with the grid, complicating the analysis. Therefore for such a mesh a Richardson Extrapolation is not directly relevant. Theoretical indicators for higher dimensional studies are scarce and one generally has to rely on pertinent numerical experiments in order to provide qualitative estimates for the error. Quon 1983, for example, tried to assess the effects of a non-uniform grid by considering the standard problem using both finite difference and finite element methods. Despite the large number of results that were presented, the conclusions of that study may be regarded as somewhat ambivient and indicate the difficulties involved. However, Quon 1983 observed that 3 or 4 points between the wall and the velocity maxima at mid-height were usually necessary for reliable predictions to be obtained, with a similar resolution for the horizontal boundary layers. Generally, some initial

Table 3. Some features of the solutions. Values in ( ) are percentage difference from available benchmark values.

Ra		$(\Delta x)^{-1}$	primitive variable		stable $\psi-\zeta$		$Benchmark$
10^3	U_{max}	20	3.623	(−0.7)	3.569	(−2.2)	3.649
		40	3.644	(−0.1)	3.619	(−0.8)	
	V_{max}	20	3.680	(−0.5)	3.594	(−2.8)	3.697
		40	3.686	(−0.3)	3.675	(−0.6)	
	ψ_{max}	20	1.177	(0.3)	1.170	(−0.3)	1.174
		40	1.175	(0.09)	1.173	(−0.09)	
	Nu_{mid}	20	1.125	(0.6)	1.115	(−0.3)	1.118
		40	1.122	(0.4)	1.117	(−0.09)	
10^4	U_{max}	20	15.891	(−1.8)	15.78	(−2.5)	16.178
		40	16.064	(−0.7)	16.09	(−0.5)	
	V_{max}	20	19.307	(−1.6)	18.47	(−5.9)	19.617
		40	19.517	(−0.5)	19.42	(−1.0)	
	ψ_{max}	20	5.023	(−0.9)	5.088	(0.3)	5.071
		40	5.058	(−0.3)	5.070	(−0.02)	
	Nu_{mid}	20	2.307	(2.9)	2.213	(−1.3)	2.243
		40	2.283	(1.8)	2.235	(−0.4)	
10^5	U_{max}	20	32.654	(−5.6)	34.97	(0.7)	34.73
		40	34.087	(−1.9)	34.51	(−0.6)	
		80	34.575	(−0.4)	34.62	(−0.3)	
	V_{max}	20	65.060	(−5.1)	57.68	(−15.9)	68.59
		40	67.816	(−1.1)	66.28	(−3.4)	
		80	68.330	(−0.4)	67.95	(−0.9)	
	ψ_{max}	20	9.406	(−2.1)	10.00	(4.0)	9.612
		40	9.525	(−0.9)	9.657	(0.5)	
		80	9.590	(−0.2)	9.620	(0.08)	
	Nu_{mid}	20	4.735	(4.8)	4.507	(−0.3)	4.519
		40	4.611	(2.0)	4.508	(−0.2)	
		80	4.569	(1.1)	4.516	(−0.07)	
10^6	U_{max}	40	62.436	(−3.4)	65.65	(1.6)	64.63
		80	63.936	(−1.1)	64.70	(0.1)	
	V_{max}	40	214.05	(−2.4)	200.3	(−8.7)	219.36
		80	214.53	(−2.2)	216.1	(−1.5)	
	ψ_{max}	40	16.59	(−1.0)	17.46	(4.2)	16.750
		80	16.70	(−0.3)	16.91	(1.0)	
	Nu_{mid}	40	9.071	(3.1)	8.951	(1.7)	8.799
		80	8.916	(1.3)	8.840	(0.5)	
	$Error$	max		(5.6)		(15.9)	
		min		(0.09)		(0.02)	

Table 4. A comparison of Uniform and Non-uniform 21 x 21 grids at $Ra = 10^6$. Values in ( ) are percentage difference from available benchmark values.

	Uniform		Non-uniform		*Benchmark*
U_{max}	58.93	(−8.8)	61.93	(−4.2)	64.63
V_{max}	212.2	(−3.3)	215.4	(−1.8)	219.4
ψ_{mid}	16.26	(−0.4)	16.06	(−1.6)	16.32
ψ_{max}	16.58	(−1.0)	16.44	(−1.9)	16.75
Nu_{mid}	9.835	(11.8)	8.974	(2.0)	8.799
Nu_0	12.38	(40.4)	9.015	(2.2)	8.817
Error	*max*	(40.4)		(4.2)	
	min	(0.4)		(1.6)	
$(\Delta x_{min})^{-1}$	20		64		
CPU(sec)	18.9		15.9		

computations with different grids will have to be performed to ensure a proper choice of mesh for accurate solutions.

In order to demonstrate the utility of non-uniform grids, results for $Ra = 10^6$ and a 21 x 21 grid are now discussed. Initially a uniform grid was used. The results for this grid (see Table 4) were not included in Table 3 because it is evidently too coarse to resolve the thin vertical boundary layers. Two additional indicators from the available benchmark (de Vahl Davis 1983) have been included in Table 4 in order to better assess the overall quality of solutions obtained.

The non-uniform grid with which comparisons have been made is shown in Figure 4. Note that the grid is non-uniform in the direction normal to the thermally active walls only. The mesh is initially uniformly spaced near the vertical walls and the local mesh expansion ratio in x direction $[hx(m+1)/hx(m)]$ is successively increased to a maximum of 1.5 near the vertical symmetry plane. The smallest grid width was $\Delta x \simeq 1/64$.

The data obtained via the non-uniform grid shows a significant improvement over the uniform case, particularly the heat transfer (see Table 4). With the exception of the stream function, the non-uniform distribution yields results which are comparable to a 41 x 41 uniform grid. This is achieved partly because increased resolution of the boundary layer flow also enforces a better integral prediction of the cross-flow (via the mass constraint) even if the mesh in the vertical direction remains unchanged.

Figure 5 illustrates clearly the improved prediction of v at mid-height. Surprisingly, improvement is maintained beyond the outer edge, where one would expect the large mesh expansion ratios to result in increased truncation error. In Figure 6 the u velocity maximum has improved significantly although its location has contracted slightly towards the wall.

Another advantage is that the non-uniform grid solutions are about 15% cheaper to obtain in this case.

Finally, some comments regarding the stream function in the core are necessary. As Table 4 indicates, the non-uniform grid predicted slightly worse values for the core and maximum stream function. These differences can be attributed in part to the fact that it is a derived variable in the present formulation, and the accuracy of the integration used to obtain it influences the final result (Behnia et al 1988).

Nevertheless, one cannot expect the core region to be properly resolved by the present method. This may present some difficulties for the fully enclosed standard problem because of the intimate coupling between the boundary layer and the core regions. The core is fully enclosed by the boundary layers and is dependent upon them. The boundary layers in turn are influenced by the core, and their mutual interaction remains an unsolved problem in all confined convection configurations (Ostrach 1982b, 1988).

Flow subregions, such as cells and layers, may be imbedded in the core and may not be revealed, and it is implicit in the success of the present method that the presence of these has only a low order influence on the gross features of the flow (e.g. heat transfer).

6 Acknowledgements

Financial support for this work was provided by the Australian Research Grants Scheme. Computing resources were provided by a joint IBM-UNSW project.

7 References

Anderson,D.A., Tannehill,J.C. and Pletcher,R.H. 1984 *Computational Fluid Mechanics and Heat Transfer* (Hemisphere).

Batchelor,G.K. 1954 Heat Transfer by Free Convection across a Closed Cavity between Vertical Boundaries at Different Temperatures *Quart. Appl. Math.* vol 12, pp209-233.

Behnia,M., Stella,F., de Vahl Davis,G. and Guj,G. 1988 Natural Convection in a Cavity: A Comparison of Different Numerical Methods *Proc. ENCIT88* Brazil, pp321-324.

Behnia,M., Wolfshtein,M. and de Vahl Davis,G. 1990 A Stable Fast Marching Scheme *Int. J. Num. Meth. Fluids* vol 10, pp607-621.

Brian,P.L.T. 1961 A Finite-Difference Method of High-order Accuracy for the Solution of Three-Dimensional Transient Heat Conduction Problems *AIChE J.* vol 47, pp367-370.

Briley,W.R. 1974 Numerical Method for Predicting Three Dimensional Steady Viscous Flow in Ducts *J. Comp. Phys.* vol 14, pp8-28.

Catton,I. 1978 Natural Convection in Enclosures *Proc. 6th Int. Heat Transfer Conf.* Toronto, vol 6, pp13-31.

de Vahl Davis,G. 1976 Program FRECON for the Numerical Solution of Free Convection in a Rectangular Cavity *Univ. NSW School of Mech. and Ind. Eng. Rept. 1976/FMT/1.*

de Vahl Davis,G. 1983 Natural Convection in a Square Cavity: A Bench Mark Numerical Solution *Int. J. Num. Meth. Fluids* vol 3, pp249-264.

de Vahl Davis,G. 1986 Finite Difference Methods for Natural and Mixed Convection in Enclosures *Proc. 8th Int. Heat Transfer Conf.* San Francisco, vol 1, pp101-109.

de Vahl Davis,G., and Jones,I.P. 1983 Natural Convection in a Square Cavity: A Comparison Exersize *Int. J. Num. Meth. Fluids* vol 3, pp227-248.

Douglas,J.Jr 1962 Alternating Direction Methods for Three Space Variables *Numer. Math.* vol 4, pp41-63.

Eckert,E.R.G., and Carlson,W.G. 1961 Natural Convection in an Air Layer Enclosed Between Two Vertical Planes with Different Temperatures *Int. J. Heat Mass Transfer* vol 2, pp106-120.

Elder,J.W. 1965a Laminar Free Convection in a Vertical Slot *J.Fluid Mech.* vol 23, pp77-98.

Elder,J.W. 1965b Turbulent Free Convection in a Vertical Slot *J.Fluid Mech.* vol 23, pp99-111.

Gebhart,B., Jaluria,Y., Mahajan,R.L. and Sammakia,B. 1988 *Buoyancy-Induced Flows and Transport* (Hemisphere).

Harlow,F.H. and Welsh,J.E. 1965 Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface *Phys. Fluids* vol 8, pp2182-2189.

Hoogendoorn,C.J. 1986 Natural Convection in Enclosures *Proc. 8th Int. Heat Transfer Conf.* San Francisco, vol 1, pp111-120.

Isenberg,J. and de Vahl Davis,G. 1975 Finite Difference Methods in Heat and Mass Transfer, in *Topics in Transport Phenomena* ed. C.Gutfinger (Wiley) pp457-553.

Kreiss,H.O., Manteuffel,T.A., Swartz,B., Wendroff,B. and White,A.B.Jr 1986 Supra-Convergent Schemes on Irregular Grids *Math. Comput.* vol 47, pp537-554.

Lankhorst,A.M., Angirasa,D. and Hoogendoorn,C.J. 1989 Experimental Investigation of Buoyancy Induced Flows in Enclosures at High Rayleigh Numbers, in *Eurotherm seminar No.11* "Natural Convection in Enclosures" (Harwell Laboratories) pp95-97.

Mallinson,G.D. and de Vahl Davis,G. 1973 The Method of False Transient for the Solution of Coupled Elliptic Equations *J. Comp. Phys.* vol 12, pp435-461.

Manteuffel,T.A. and White,A.B.Jr 1986 The Numerical Solution of Second Order Boundary Value Problems on Non-uniform Meshes *Math. Comput.* vol 47, pp511-536.

Ostrach,S. 1982a Low-gravity Fluid Flows *Ann. Rev. Fluid Mech.* vol 14, pp313-345.

Ostrach,S. 1982b Natural Convection Heat Transfer in Cavities and Cells *Proc. 7th Int. Heat Transfer Conf.* München, vol 1, pp365-379.

Ostrach,S. 1988 Natural Convection in Enclosures *J. Heat Transfer* vol 110, pp1175-1190.

Patankar,S.V. 1980 *Numerical Heat Transfer and Fluid Flow* (Hemisphere).

Patankar,S.V. and Spalding,D.B. 1972 A Calculation Procedure for Heat, Mass and Momentum Transfer in Three-Dimensional Parabolic Flows *Int. J. Heat Mass Transfer* vol 15, pp1787-1806.

Peyret,R. and Taylor,T.D. 1983 *Computational Methods For Fluid Flow* (Springer-Verlag).

Piacsek,S.A. and Williams,G.P. 1970 Conservation Properties of Convection Difference Schemes *J. Comp. Phys.* vol 6, pp392-405.

Quon,C. 1983 Effects of Grid Distribution on the Computation of High Rayleigh Number Convection in a Differentially Heated Cavity, in *Numerical Properties and Methodologies in Heat Transfer* ed. T.M. Shih (Hemisphere) pp261-281.

Reizes,J.A.,Leonardi,E. and de Vahl Davis,G. 1984 Problems with Derived Variable Methods for the Numerical Solution of Three-Dimensional Flows, in *Computational Techniques and Applications:CTAC-1983* ed. J.Noye and C.A.Fletcher (North Holland) pp903-913.

Roache,P.J. 1972 *Computational Fluid Dynamics* (Hermosa).

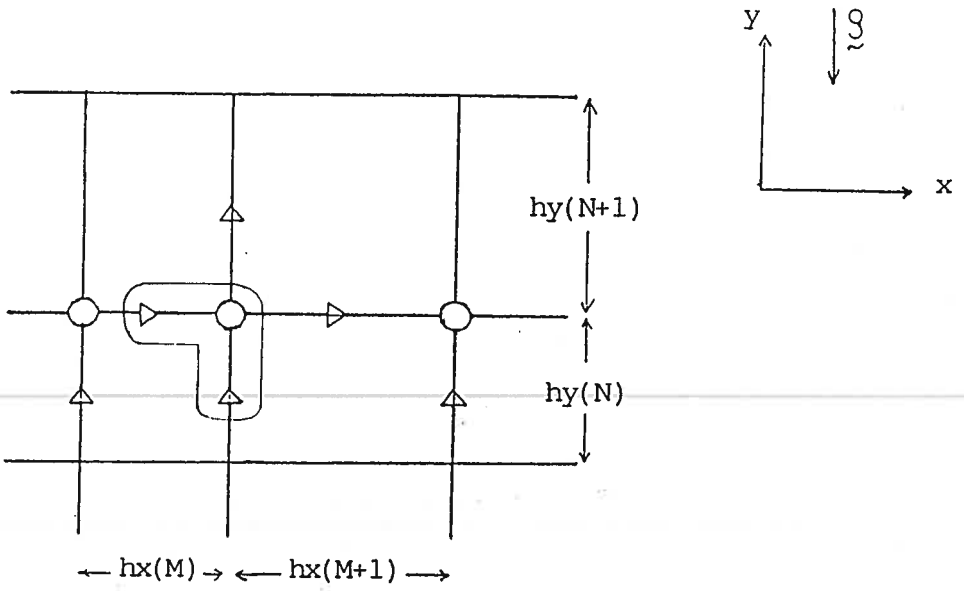
Samarskii,A.A. and Andreyev,V.B. 1963 On a High-Accuracy Difference Scheme for an Elliptic Equation with Several Space Variables *USSR Comp. Math. and Math. Phys.* vol 3, pp1373-1382.

Schumann,U. and Sweet,R.A. 1976 A Direct Method for the Solution of Poisson's Equation with Neumann Boundary Conditions on a Staggered Grid of Arbitrary Size *J. Comp. Phys.* vol 20, pp171-182.

Simpkins,P.G. 1987 Convection in Laterally Heated Cavities, in *Computational Fluid Dynamics* ed. G.de Vahl Davis and C.Fletcher (North-holland) pp157-173.

Turner, J.S. 1973 *Buoyancy Effects in Fluids* (Cambridge University Press).

Wong,A.K. and Reizes,J.A. 1986 The Vector Potential in the Numerical Solution of Three-Dimensional Fluid Dynamics Problems in Multiply-Connected Regions *J. Comp. Phys.* vol 55, pp98-114.



○ = unstaggered variables T, P, ϕ
▷ = staggered u velocity
△ = staggered v velocity
⌜⌝ = computational cell containing variables with same subscripts ($u_{m,n}, T_{m,n}$ etc)

Figure 1: Definition of grid spacing and computed variables on a non-uniform staggered grid.

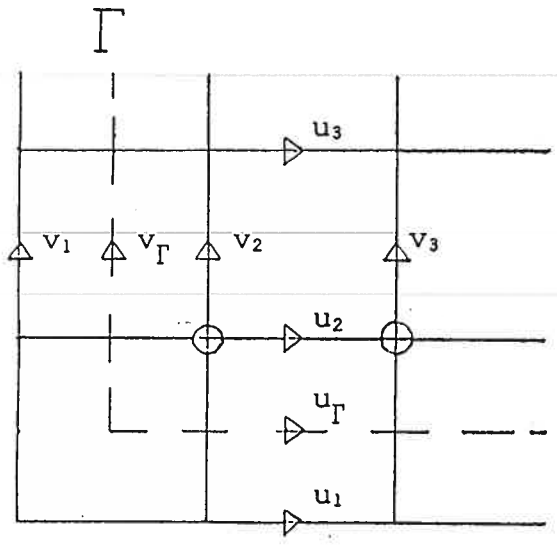


Figure 2: Staggered grid near solid boundary (Γ).

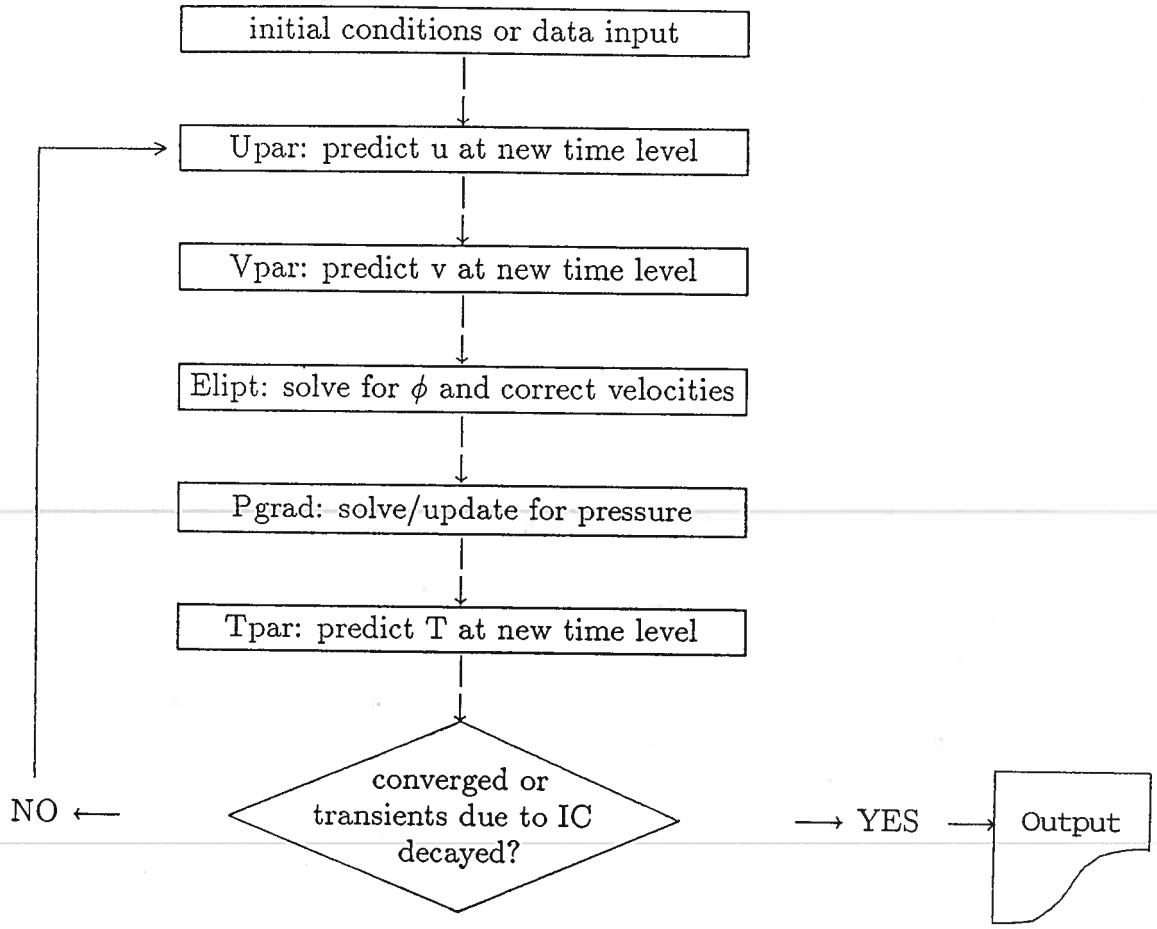


Figure 3: Block diagram of the solution procedure.

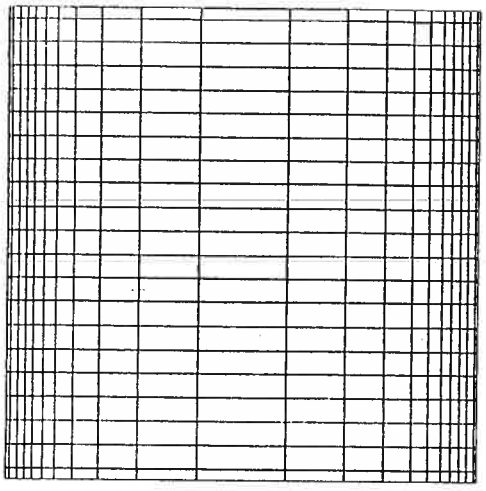


Figure 4: A non-uniform 21×21 grid (see text).

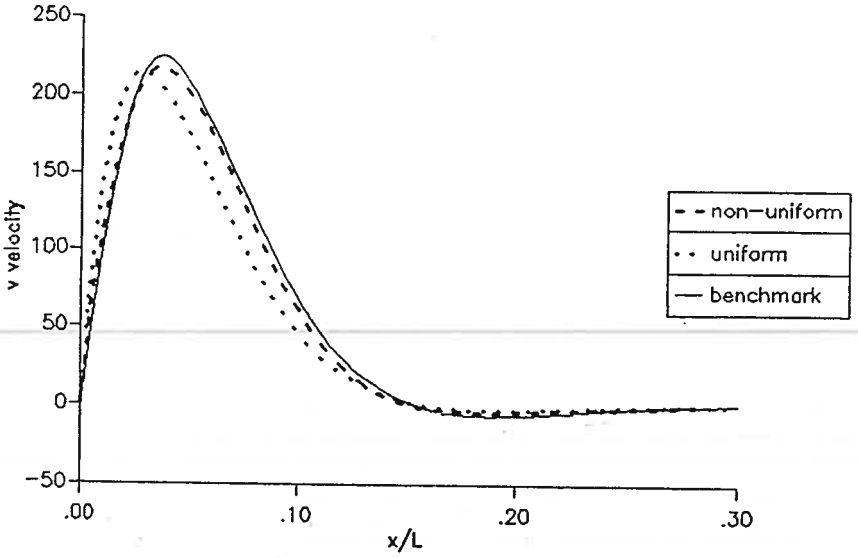


Figure 5: Vertical velocity distribution normal to the wall at midheight ($y/H = 0.5$).

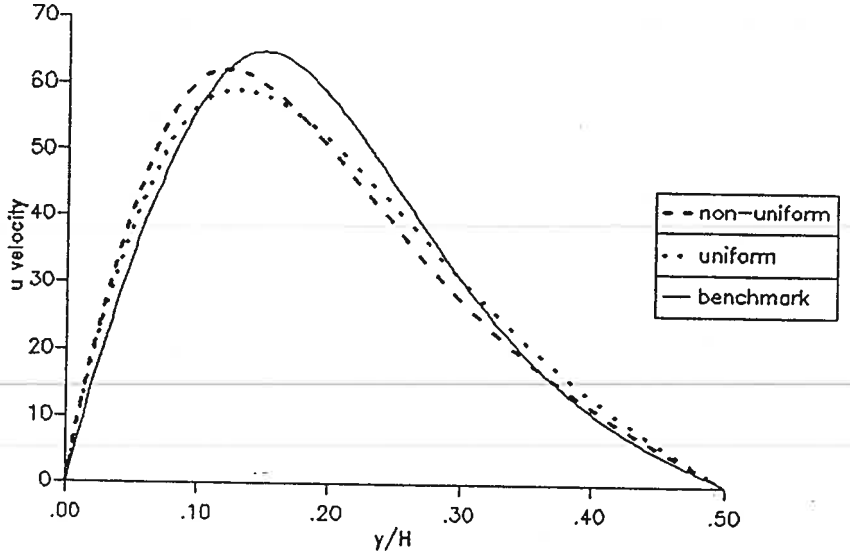


Figure 6: Horizontal velocity distribution normal to the wall at midwidth ($x/L = 0.5$).